Theoretical Computer Science 411 (2010) 783–798

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On the complexity of kings

Edith Hemaspaandra [a], Lane A. Hemaspaandra [b], Till Tantau [c,*], Osamu Watanabe [d]

[a] Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623, USA
[b] Department of Computer Science, University of Rochester, Rochester, NY 14627, USA
[c] Institut für Theoretische Informatik, Universität zu Lübeck, D-23538 Lübeck, Germany
[d] Department of Mathematical & Computing Sciences, Tokyo Institute of Technology, Tokyo, 152-8552, Japan

ARTICLE INFO

Keywords:
Diameter
Kings
Polynomial hierarchy
Succinct representations
Tournaments

ABSTRACT

A king in a directed graph is a vertex from which each vertex in the graph can be reached through paths of length at most two. There is a broad literature on tournaments (completely oriented digraphs), and it has been known for more than half a century that all tournaments have at least one king. Recently, kings have proven useful in theoretical computer science, in particular in the study of the complexity of reachability problems and semifeasible sets.

In this article, we study the complexity of recognizing kings. For each succinctly specified family of tournaments, the king problem is already known to belong to Π_2^{p}. We prove that the complexity of kingship problems is a rich enough vocabulary to pinpoint every nontrivial many-one degree in Π_2^{p}. That is, we show that *every* set in Π_2^{p} other than $\emptyset$ and Σ^* is equivalent to a king problem under $\leq_{\mathrm{m}}^{\mathrm{p}}$-reductions. Indeed, we show that the equivalence can even be realized by relatively simple padding, and holds even if the notion of kings is redefined to refer to k-kings (for any fixed $k \geq 2$)—vertices from which all vertices can be reached through paths of length at most k. In contrast, we prove that for each succinctly specified family of tournaments the source problem (the problem of deciding whether a given vertex v has the property that there exists a k such that v is a k-king) also falls within Π_2^{p}, yet cannot be Π_2^{p}-complete—or even NP-hard—unless $\mathrm{P} = \mathrm{NP}$.

Using these and related techniques, we obtain a broad range of additional results about the complexity of king problems, diameter problems, and radius problems. It follows easily from our proof approach that the problem of testing kingship in succinctly specified graphs (which need not be tournaments) is Π_2^{p}-complete. We show that the radius problem for arbitrary succinctly represented graphs is Σ_3^{p}-complete, but that in contrast the diameter problem for arbitrary succinctly represented graphs (or even tournaments) is Π_2^{p}-complete.

1. Introduction

1.1. Problem statement and main structural result

In this article, we study the complexity of recognizing kings and k-kings in graphs. For each $k \geq 0$, a *k-king* of a graph is a vertex such that every vertex can be reached from it through a path of length at most k. By convention, a vertex of a

* Corresponding author.
E-mail address: tantau@tcs.uni-luebeck.de (T. Tantau).
URLs: http://www.cs.rit.edu/~eh/ (E. Hemaspaandra), http://www.cs.rochester.edu/u/lane/ (L.A. Hemaspaandra), http://www.tcs.uni-luebeck.de/mitarbeiter/tantau/ (T. Tantau), http://www.is.titech.ac.jp/~watanabe/ (O. Watanabe).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2009.10.015

graph is said to be a *king* if every vertex of the graph can be reached from it through a path of length at most two. So "king" and "2-king"—and "kingship" and "2-kingship"—are synonymous in this article, as they are also in the general literature on kings.

In the *k-kingship problem*, we are given a graph and a vertex as inputs and would like to tell whether the vertex is a k-king. We can vary the problem by allowing different ways of encoding graphs (the more succinctly, the harder the problem) and by allowing different kinds of input graphs (the more restricted, the easier the problem).

Much is known about the *existence* of kings in graphs. For example, in the 1950s Landau [10] discovered the simple but lovely result that every tournament has a king. A tournament is a directed graph G such that for each pair u and v of distinct vertices exactly one of the directed edges $u \to v$ or $v \to u$ is present in the graph and such that there are no loops. A well-known way (see [22]) to easily see that Landau's result holds is to note that every vertex with maximum out-degree must be a king. More recently, similar results were proven for generalizations of tournaments, such as multipartite tournaments ([5, 13], see also [1] and the references therein).

When graphs are specified explicitly in the natural way (say, by an adjacency matrix), it is not hard to see that for each $k \geq 0$ the k-kingship problem is first-order definable and thus very simple from a computational point of view. However, when we specify graphs *succinctly*, for each $k \geq 1$ the complexity of k-kingship problems jumps from having an upper bound of "first-order definable" to having a lower bound of "coNP-hard" (or worse). There are different ways of specifying graphs succinctly, ranging from the general Galperin–Wigderson model to the polynomial-time uniform tournament family specifiers that arise in the study of semifeasible sets.

In the Galperin–Wigderson model, input graphs are specified as follows: A directed graph G with a vertex set $\{0, 1\}^n$ is specified using a circuit C with $2n$ input gates and one output gate. For any two vertices $x, y \in \{0, 1\}^n$, there is an edge $x \to y$ in G if and only if $C(xy) = 1$. (This definition does allow the possibility of self-loops.) Note that a circuit whose size is polynomial in n can encode a graph whose vertex set has size 2^n, which is exponential in n. For this model, the k-kingship problem can be formalized as follows:

$$\text{SUCCINCT-}k\text{-KINGS} = \{\langle \mathrm{code}(C), x\rangle \mid C \text{ specifies (in the manner specified above) a graph } G \text{ in which } x \text{ is a } k\text{-king}\}.$$

In the above definition, we used $\mathrm{code}(C)$ to denote a standard binary encoding of the circuit C. Furthermore, $\langle \mathrm{code}(C), x\rangle$ is a standard binary encoding of the circuit C paired with a bitstring x. The exact definition of the pairing function $\langle ., .\rangle$ is detailed in the preliminaries section.

In the tournament family specifier model, input graphs are specified using *polynomial-time computable, commutative selector functions*. A *selector function* f gets two words u and v as inputs and outputs one of them, thereby telling us where the edge between u and v heads. More formally, a selector function $f : \{0, 1\}^* \times \{0, 1\}^* \to \{0, 1\}^*$ defines an (infinite) graph with the vertex set $\{0, 1\}^*$ where there is an edge from u to v if and only if $f(u, v) = v$ and $u \neq v$. The graph will be a tournament if f is commutative, that is, if for each u and v it holds that $f(u, v) = f(v, u)$. Polynomial-time selector functions were originally introduced by Selman [16–18] in his study of the so-called P-selective sets: P-selective set can be defined as the sets of vertices in tournaments specified by polynomial-time computable commutative selector functions that are closed under reachability.

Instead of using a selector function to describe a single infinite tournament, we can also use them to describe one tournament per word length: Given a commutative selector function f and a word length n, we say that f describes the length-n tournament whose vertex set is $\{0, 1\}^n$ and whose edge set is defined as above: There is an edge from $u \in \{0, 1\}^n$ to $v \in \{0, 1\}^n$ if $f(u, v) = v$ and $u \neq v$. In this model, we can also consider the k-kingship problem:

$$k\text{-Kings}_f = \{x \in \{0, 1\}^* \mid x \text{ is a } k\text{-king in the length-}|x| \text{ tournament specified by } f\}.$$

One can view the k-Kings$_f$ problems (one for each f) as very restricted cases of the more general SUCCINCT-k-KINGS problem: For k-Kings$_f$ we must check k-kingship for *a single tournament per word length.* In complexity-theoretic terms, this tremendous uniformity of specification—a polynomial-time computable function specifying for us a single tournament at each length—will naturally tend to "tie our hands" in terms of showing hardness for higher levels of the polynomial hierarchy. Nonetheless, we show that we can free our hands from those cords: The core of this article is a proof showing that the one-tournament-per-length has such potential structural richness that asking about kingship of different vertices within that structure allows us to achieve any desired hardness level inside Π_2^{p}. In particular, Corollary 4.2 shows that there is a selector-specified kings problem that is Π_2^{p}-complete for some fixed selector function.

A language $L \subseteq \{0, 1\}^*$ for which there exists a commutative polynomial-time selector function f such that $L = k$-Kings$_f$ will be called a P-*k-king language*. Our main interest in this article is to study which languages are P-k-king languages. Our main structural result is that, for each $k \geq 2$, every language in $\Pi_2^{\mathrm{p}} - \{\emptyset, \Sigma^*\}$ is many-one equivalent to a P-k-king language. Informally put, this shows that k-kings languages are comprehensively descriptive in terms of naming the complexity of the nontrivial Π_2^{p} many-one degrees.

We obtain this result by an even stronger tool, which shows something about the uniformity and simplicity of a set of reductions that can instantiate the above equivalences. Namely, we show that, for every $k \geq 2$, a language L is in Π_2^{p} if and only if $\mathrm{pad}'_j(L)$ is a P-k-king language for some j. Here, pad'_j is a padding operator whose exact definition will be given later.

1.2. Motivations for studying P-k-king languages

Our study of P-k-king languages is motivated from several contrasting directions.

Relationship to the radius problem. Kings (recall: as is standard, by this we mean 2-kings) and k-kings are closely related to the *radius problem* for graphs. A ball of radius r around a vertex v is the set of vertices that can be reached from v within r steps. The radius of a graph is the smallest radius of a ball that covers the whole graph. This means that the radius of a graph is at most r if and only if there *exists* an r-king in the graph. (Note, in contrast, that the k-king problems focus on whether a given vertex, which is explicitly stated as part of the input, is a k-king.)

We use our results on P-k-king languages to give a short proof that radius problems for succinctly specified graphs (using the Galperin–Wigderson model) are complete for the *third* level of the polynomial hierarchy [11,19], i.e., are complete for $\Sigma_3^{\mathrm{p}} = \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}}$. This result is interesting in its own right. While for the first level of the polynomial hierarchy (NP) countless natural complete problems are known, for higher levels the collection of such problems is less extensive (see also Section 1.3's comments on complete sets for such classes). The succinct radius problem is a new and fairly natural problem that is complete for Σ_3^{p}.

Relationship to the diameter problem. Kings and k-kings are also closely related to the *diameter problem* for graphs. The *diameter* of a graph is the smallest number d such that for every two vertices there is a directed path from the first to the second vertex of length at most d; if the graph is not strongly connected, the diameter is infinite. This means that a graph has diameter at most d if and only if *every* vertex of the graph is a d-king of the graph.

Based on this relationship, we show that the diameter problem for succinctly represented graphs are complete for $\Pi_2^{\mathrm{p}} = \mathrm{coNP}^{\mathrm{NP}}$.

Relationship to P-selective sets. P-2-king languages are closely related to P-selective languages. For a P-selective language A, for each n, within the length-n graph specified by the selector function it always holds that the reachability closure of the length n words of A is precisely the length n words of A. For a P-2-king language, the words in the language of length n are the kings in the length-n tournament specified by the selector function. This means that for a P-selective set the kings of the tournaments induced by a selector are (speaking very informally) the "least likely" words to be contained in the language. More precisely, unless all words of a given word length are in the language, none of the kings of the tournament specified by the selector for this word length is in the language. This observation can be used to show that P-selective sets cannot be $\Pi_2^{\mathrm{p}}/1$-immune [9]. Finally, we mention that tournaments have proven useful in many areas in addition to the study of P-selective sets. For example, tournaments are important in social choice theory, e.g., the computationally important election systems of Llull and Copeland are tournament-based (see [2]).

Relationship to the second level of the polynomial hierarchy. Despite the close relationship of P-2-king languages and P-selective languages, there are fundamental differences. For example, it is easy to see that all P-k-king languages are in Π_2^{p}, see [8] for a detailed proof, but it is well known that P-selective languages can be "arbitrarily complex" in a sense that can be crisply formalized (for example, Selman's seminal articles on P-selectivity established that for every tally language A there is a P-selective set that is $\leq_{\mathrm{T}}^{\mathrm{p}}$-equivalent to A). This encourages us to investigate which languages are P-k-king languages. Many languages in Π_2^{p} are not P-k-king languages—for example, since every tournament has a king, a P-2-king language contains at least one word for every word length. However, the tool underpinning our structural results shows that for every $k \geq 2$ and every language $L \in \Pi_2^{\mathrm{p}}$ a certain padded version of L is a P-k-king language. Thus, although not every language in Π_2^{p} is a P-k-king language, for every such language a closely related language is a P-k-king language. And, from this we have our main structural result, which is that every Π_2^{p} (many-one) degree, except those of $\emptyset$ and Σ^*, contains a P-k-king language. In fact, something even stronger than many-one degrees holds, due to the the precise proof we use to show this. We, in fact, establish that for every $k \geq 2$ every language in Π_2^{p} except $\emptyset$ and Σ^* is equivalent to a P-k-king language even under first-order reductions. In particular, for every $k \geq 2$ there exist P-k-king languages that are complete for Π_2^{p} with respect to first-order reductions.

Relationship to quantifier characterizations. The quantifier characterization of the polynomial hierarchy states that a language L is in Π_2^{p} if and only if there exist a polynomial p and a ternary polynomial-time decidable relation R such that $x \in L \iff (\forall y \in \{0,1\}^{p(|x|)})(\exists z \in \{0,1\}^{p(|x|)})[R(x,y,z)]$. For P-2-king languages a more restrictive characterization is possible: A language L is a P-2-king language if and only if there exists a binary polynomial-time decidable relation S such that $x \in L \iff (\forall y \in \{0,1\}^{|x|})(\exists z \in \{0,1\}^{|x|})[S(x,y) \wedge S(y,z)]$ and such that for all distinct $x, y \in \{0,1\}^*$ we have $S(x,y) \leftrightarrow \neg S(y,x)$.

Relationship to the top Toda equivalence classes of tournaments. Kings and k-kings can be used in the study of the top Toda equivalence classes of tournaments. Given a commutative P-selector f and a word length n, two elements of the length-n tournament induced by f are said to be *Toda equivalent* if there is a path from the first to the second element and also a path from the second to the first element. For each word length there exists what is called a *top Toda equivalence class* [8], by which we mean the unique equivalence class (in that tournament) whose members are not pointed to by members of any other equivalence class of that tournament. A different way of phrasing this is as follows: The top Toda equivalence class of the length-n tournament is the unique strongly connected component from which all other vertices can be reached.

Note that a vertex v will belong to the top Toda equivalence class precisely if all vertices in the tournament are reachable from v. So, asking whether a vertex belongs to the top Toda equivalence class is the same as asking whether the vertex has the property that there exists a k for which the vertex is a k-king. Analogously to P-k-king languages, we can define *Top-Toda languages* for commutative P-selectors f:

$$\text{Top-Toda}_f = \{x \in \{0, 1\}^* \mid \text{there exists a } k \text{ such that } x \text{ is a } k\text{-king in the length-}|x| \text{ tournament specified by } f\}.$$

At first sight, these languages might appear to be more difficult than P-k-king languages: Instead of checking whether there is a path of length at most k from the input to each vertex in the tournament, we must ask the same question for the richer universe of all paths (i. e., without an upper bound of k being imposed on their lengths). However, we prove that all these languages are also in Π_2^p. Moreover, we prove that none of these languages can be Π_2^p-complete—or even NP-hard—unless P = NP. Comparing this with P-k-king languages, which a corollary to our main structural result shows *are* sometimes Π_2^p-complete, we see that all Top-Toda languages are seemingly *easier* than some P-k-king languages.

1.3. Related work

The work most closely related to that of this article is the work of Nickelsen and Tantau on the complexity of reachability problems [12], the path-breaking modeling and complexity work of Galperin and Wigderson [3], and the existing work on the complexity of kings and in particular their use in the study of the semifeasible sets [7,9,8].

It is well worth mentioning that without the seminal work of Landau [10], which showed that kings always exist in tournaments, it is unlikely that the notion of kings would even be available for study. And, Landau's work has led to a rich (though, naturally, not complexity-theoretic) body of work on the existence of kings or k-kings in a variety graph-theoretic structures (for example, for the case of multipartite tournaments see [5,13,1] and the references therein).

For reasons of focus and coherence, all tournaments in this article follow the typical notion of a tournament. However, we mention in passing that one central result of this article, our Π_2^p-completeness result for the k-kings problem, has been studied for the case of j-partite tournaments (though in a more circuit-focused model) in the June 2005 technical report version of this article (available at `arXiv.org`), where a dichotomy theorem is given that completely characterizes what happens in that case, namely, for the boundary case of "1-kingship" one gets P-algorithms and for all other cases Π_2^p-completeness holds. We also mention that though the results of this article are fundamentally complexity-theoretic in nature, one can also study their recursion-theoretic siblings, and can obtain Kleene-hierarchy versions of the polynomial-hierarchy results presented here.

In this article, we will show problems to be complete for classes at the second and third levels of the polynomial hierarchy. These levels have nothing resembling the range and number of known, natural complete problems that NP has (see, for example, the famous compendium of Garey and Johnson [4]). Nonetheless, these levels do have a larger range and number of known, natural complete problems than many people realize. Schaefer and Umans have provided a very nice "Garey and Johnson" for classes at levels of the polynomial hierarchy beyond the first [14,15].

1.4. Organization of this article

Section 2 provides notations, definitions, and some important lemmas that we will need in the article's result sections. Section 3 studies the complexity of the diameter problem, and shows that it is complete for the Π_2^p level of the polynomial hierarchy. That section also introduces tools that will be used in subsequent proofs in the article. Section 4 proves our main structural result, namely, that k-kings problems have the descriptive flexibility to name every nontrivial Π_2^p degree. We do so by showing that for each $k \geq 2$ it holds that by a certain family of padding functions each Π_2^p language can be turned into a P-k-king language. Section 5 studies the complexity of the radius problem, and shows that it is complete for the Σ_3^p level of the polynomial hierarchy, which provides an interesting contrast with Section 3's Π_2^p-completeness result for the diameter problem. Section 6 studies Top-Toda languages and shows that although such languages are always in Π_2^p, they cannot be Π_2^p-complete (or even NP-hard), unless P = NP. Section 7 is our conclusion.

2. Basic definitions and tools

In this section, we introduce the notation and terminology used in the rest of the article. We also prove lemmas on key concepts, which will be used later in the article.

2.1. Bitstrings, alphabets, and padding

Throughout this article, $\Sigma = \{0, 1\}$. We refer to elements of $\{0, 1\}^* = \Sigma^*$ as bitstrings. The *length* of a bitstring b is denoted $|b|$. We define a pairing function $\langle ., . \rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$ as follows (unlike some other articles, we will not require our pairing function to be a surjective function): For every two bitstrings $x, y \in \Sigma^*$ where the individual bits of x are x_1 to x_n, let $\langle x_1x_2 \cdots x_n, y\rangle = 0x_10x_2 \cdots 0x_n1y$. This function, which clearly is injective, has a number of useful properties. Those that will be used in this article are listed in the following lemma, whose straightforward proof is omitted.

Lemma 2.1. *The pairing function $\langle .,. \rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$ has the following properties:*

(i) *It is polynomial-time computable.*
(ii) *It is polynomial-time invertible (its range is in* P *and there exist two polynomial-time computable functions σ_1 and σ_2 such that, given a string z in the range of $\langle .,. \rangle$, it holds that $\langle \sigma_1(z), \sigma_2(z) \rangle = z$).*
(iii) *For all words $x, y, x', y' \in \Sigma^*$ with $|x| = |x'|$ and $|y| = |y'|$, we have $|\langle x, y \rangle| = |\langle x', y' \rangle|$.*
(iv) *The range of the function does not include any word from $\{0\}^*$ (no word pair is mapped to an element of $\{0\}^*$).*

For a tuple $(x_1, \ldots, x_n)$ of words, $n \geq 1$, let $\langle x_1, \ldots, x_n \rangle = \langle x_1, \langle x_2, \ldots, \langle x_{n-1}, x_n \rangle \cdots \rangle\rangle$.
For a positive integer j, we define a padding function $\mathrm{pad}_j : \Sigma^* \to \Sigma^*$ by

$$\mathrm{pad}_j(x) = x0^{|x|^j + j + 3}.$$

Thus, we add $|x|^j + j + 3$ zeros after x. The reason for the slightly startling "$+\ 3$" will become clear later on. Note that for every word $y \in \Sigma^*$ there can be at most one word x such that $\mathrm{pad}_j(x) = y$ and, if such an x exists, it is easy to compute.

We define two padded versions of languages. The "usual" way to define a padded version of a language L is to consider the image of L under the padding function pad_j, that is, for a given language $L \subseteq \Sigma^*$ let $\mathrm{pad}_j(L) = \{\mathrm{pad}_j(x) \mid x \in L\}$. The "interesting" words in a padded language are those in $\mathrm{pad}_j(\Sigma^*)$. Words outside $\mathrm{pad}_j(\Sigma^*)$ are not in $\mathrm{pad}_j(L)$. For the second padded version of L, we change this latter property: The membership of the words in $\mathrm{pad}_j(\Sigma^*)$ is the same, but (almost) all other words are *in* the second padded version. Formally, for a language $L \subseteq \Sigma^*$ we define

$$\mathrm{pad}'_j(L) = \mathrm{pad}_j(L) \cup \big(\Sigma^* - \mathrm{pad}_j(\Sigma^*) - \{1, 11\}\big).$$

Once more, there is a startling part of the definition, namely the "$-\ \{1, 11\}$" and, again, this will be explained later on. The important properties of the padded versions of a language L are listed in the next lemma, whose straightforward proof is also omitted.

Lemma 2.2. *Let $L \subseteq \Sigma^*$ and let j be a positive integer. The language $L' = \mathrm{pad}'_j(L)$ has the following properties:*

(i) *For all words $x \in \Sigma^*$ it holds that $x \in L$ if and only if $\mathrm{pad}_j(x) \in L'$.*
(ii) *L' contains one word of length* 0*, one word of length* 1 *and three words of length* 2*.*
(iii) *For all word lengths $n \geq 3$ that are not of the form $n = m + m^j + j + 3$ for some nonnegative integer m, all words of length n are in L'.*
(iv) *For every word length n that is of the form $n = m + m^j + j + 3$ for some nonnegative integer m, all words of length n that do not end with $m^j + j + 3$ zeros are in L'.*
(v) *If $L \neq \emptyset$ and $L \neq \Sigma^*$, then L and L' are $\leq_m^p$-equivalent.*

2.2. Graphs and tournaments

A *(directed) graph* is a pair (V, E) consisting of a nonempty vertex set V together with an edge set $E \subseteq V \times V$. Instead of $(u, v) \in E$, we also write $u \to_E v$ or just $u \to v$ when E is clear from context. The *out-degree* of a vertex u in a graph (V, E) is the number of vertices v for which $u \to v$ holds. The *in-degree* is defined analogously.

A *path of length l* in a graph is sequence $v_0, v_1, \ldots, v_l$ of vertices such that $v_{i-1} \to v_i$ holds for all $i \in \{1, \ldots, l\}$. For integers $k \geq 0$, a *k-king* of a graph is a vertex v such that there is a path of length at most k from v to every other vertex. (Note that the only graph that has a 0-king is the graph consisting of a single vertex.) A 2-king is also just called a *king*. The *diameter* of a graph is the smallest number d such that for every pair $u, v \in V$ of vertices there is a path from u to v of length at most d. Note that the diameter of a graph is exactly the smallest number k such that every vertex of the graph is a k-king. If a graph has more than one strongly connected component, its diameter is ∞. The *radius* of a graph is the smallest number r such that there exists a vertex v from which there are paths of length at most r to all other vertices. Note that the radius of a graph is exactly the smallest number k such that there exists a k-king in the graph. It is possible for a graph (though, as we will see, not for a tournament) to have a radius of ∞.

A *tournament* is a directed graph such that (a) there are no self-loops, i.e., the edge relation E is irreflexive and (b) for every pair $u, v \in V$ of distinct vertices we have either $(u, v) \in E$ or $(v, u) \in E$, but not both. It is well known that any vertex of maximal out-degree in a tournament is a king of the tournament. In particular, every tournament has a king.

Except for the tournament consisting of a single vertex or of no vertices, a tournament obviously cannot have a diameter strictly less than 2. In the following, we will often need to construct tournaments that have diameter exactly 2. The following lemmas show when and how this can be done, but first we need a definition.

Definition 2.3. Let $G = (V, E)$ be a graph. Let u and v be two new vertices that are not in V. Let $G[u, v] = (V', E')$ be the following graph: $V' = V \cup \{u, v\}$ and $E' = E \cup \{(u, v)\} \cup \{(x, u) \mid x \in V\} \cup \{(v, x) \mid x \in V\}$.

The definition states that we obtain $G[u, v]$ from G by adding the vertices u and v and adding edges from all old vertices to u, adding an edge from u to v, and adding edges from v to all old vertices.

Lemma 2.4. *Let $T = (V, E)$ be a tournament and let $u, v \notin V$ be two new vertices. Let $K \subseteq V$ be the set of kings in T. Then the set of kings of $T[u, v]$ is $K \cup \{u, v\}$. In particular, if T has diameter 2, then $T[u, v]$ has diameter 2.*

Proof. To prove the lemma we need to show that u and v are kings of $T[u, v]$ and, furthermore, that a vertex $x \in V$ is a king of $T[u, v]$ if and only if x is a king in T. First, v is a king in $T[u, v]$ as there are direct edges from v to all $x \in V$ and there is a path $v \to x \to u$ in $T[u, v]$, where $x \in V$ is an arbitrary vertex. Second, u is a king in $T[u, v]$ as there is an edge $u \to v$ and for all $x \in V$ there is a path $u \to v \to x$. Third, a king k of T is also a king of $T[u, v]$: For every $x \in V$ there is a path of length at most 2 from k to x, since k is a king in T. There is an edge $k \to u$ by construction and there is the path $k \to u \to v$. Fourth, if $x \in V$ is not a king of T, it is also not a king of $T[u, v]$: Let $y \in V$ be a vertex at a distance (measured in T) from k of at least 3. Then the distance from x to y in $T[u, v]$ is 3, but not less: There is a path $x \to u \to v \to y$, but neither the path $x \to u \to y$ nor the path $x \to v \to y$ exist. $\square$

Lemma 2.5. *There is no 4-vertex tournament of diameter 2.*

Proof. For the sake of contradiction, assume that a 4-vertex tournament T with diameter 2 exists. Clearly, no vertex can have out-degree 0 or 3. Then, all four vertices have out-degree 1 or 2. The out-degrees must sum up to 6 as there are six edges, and thus two vertices must have out-degree 1 and the other two have out-degree 2. Let u be a vertex of out-degree 1 and let v be the single vertex for which there is an edge $u \to v$ in T. We must be able to reach the other two vertices, call them z and w, within two steps from u, and thus there must be edges from v to both z and w. Consider the edge between w and z. If it goes from z to w, then the shortest path from w to z is (w, u, v, z); if it goes from w to z, then the shortest path from z to w is (z, u, v, w). In either case, we have a contradiction. $\square$

Definition 2.6. We define a 6-vertex tournament $T_6^{\text{diam}=2}$ as follows. Let $V = C \cup D$, where $C = \{c_0, c_1, c_2\}$ and $D = \{d_0, d_1, d_2\}$. We connect the vertices c_i in a "clockwise" fashion and the vertices d_i in a "counter-clockwise" fashion, that is, we add edges $c_0 \to c_1 \to c_2 \to c_0$ and $d_2 \to d_1 \to d_0 \to d_2$. We call C and D the two *cycles* of the tournament. Next, we add an edge $c_i \to d_i$ for all $i \in \{0, 1, 2\}$. In the other direction, we add an edge $d_i \to c_{(i+1) \pmod 3}$ for all $i \in \{0, 1, 2\}$. The remaining missing edges are added in any fixed, arbitrary manner.

Lemma 2.7. *The tournament $T_6^{\text{diam}=2}$ has the following properties:*

(i) *Its diameter is* 2.
(ii) *For every vertex $c \in C$ there is a vertex in $d \in D$ such that $d \to c$.*
(iii) *For every vertex $d \in D$ there is a vertex in $c \in C$ such that $c \to d$.*

Proof. For the first claim, first note that we can reach all vertices from vertex c_0 through the following paths: $c_0 \to c_1$, $c_0 \to c_1 \to c_2$, $c_0 \to d_0$, $c_0 \to c_1 \to d_1$, and $c_0 \to d_0 \to d_2$. Second, observe that the situation is symmetric for all other vertices—only the numbering is changed. For the second claim, we can reach c_i from $d_{(i+2) \pmod 3}$; for the third claim, we can reach d_i from c_i. $\square$

Lemma 2.8. *Let n be a positive integer. Then, there exists an n-vertex tournament of diameter* 2 *if and only if $n \notin \{1, 2, 4\}$.*

Proof. Let $\{x_1, \ldots, x_n\}$ be a set of vertices. We show how to construct a diameter-2 tournament $T_n^{\text{diam}=2}$ on this set for $n \notin \{1, 2, 4\}$ and show that no such tournament exists for $n \in \{1, 2, 4\}$.

For $n = 1$, the only tournament contains a single vertex and has diameter 0.

For $n = 2$, the only tournament is a single edge, and thus it has an infinite diameter.

For $n = 3$, the cycle $x_1 \to x_2 \to x_3 \to x_1$ is a tournament of diameter 2.

For $n = 4$, no tournament of diameter 2 exists by Lemma 2.5.

For $n = 5$ and all larger odd numbers, we can construct an n-vertex tournament of diameter 2 by repeatedly applying Lemma 2.4 to the 3-vertex tournament of diameter 2. When we apply Lemma 2.4 for the first time, we set $u = x_4$ and $v = x_5$. When we apply it for the second time, we set $u = x_6$ and $v = x_7$. More generally, each time we apply the lemma we set u to be x_i for an even i and we set v to be x_{i+1}.

For $n = 6$, the tournament $T_6^{\text{diam}=2}$ from Definition 2.6 has diameter 2 by Lemma 2.7.

For $n = 8$ and all larger even numbers, we construct an n-vertex tournament of diameter 2 by repeatedly applying Lemma 2.4 to the 6-vertex tournament of diameter 2. Each time we apply the lemma we set u to be x_i for an odd i and we set v to be x_{i+1}. $\square$

Definition 2.9. Given a list $(x_1, \ldots, x_n)$ of vertices for $n \geq 5$, let $T_n^{\text{diam}=2}(x_1, \ldots, x_n)$ be the tournament constructed in Lemma 2.8. When the list is clear from context, we just write $T_n^{\text{diam}=2}$.

2.3. Polynomial hierarchy

The polynomial hierarchy [11,19] can be defined in various ways that yield the same classes. However, for the purpose of this article, it is easiest to use a quantifier-based definition (see [20,23]). We say that a binary relation $R \subseteq \Sigma^* \times \Sigma^*$ is *polynomial-time decidable* if the set $\{\langle x, y\rangle \mid (x, y) \in R\}$ is in P. We define polynomial-time decidable ternary or quaternary relations similarly. With this definition, we can define the two levels of the polynomial hierarchy that will be of interest to us, namely Π_2^{p} and Σ_3^{p}, as follows: A language L is in Π_2^{p} if and only if there exist a polynomial p and a polynomial-time decidable relation $R \subseteq \Sigma^* \times \Sigma^* \times \Sigma^*$ such that for all words $x \in \Sigma^*$ we have

$$x \in L \iff (\forall y \in \Sigma^{p(|x|)})(\exists z \in \Sigma^{p(|x|)})[R(x, y, z)]. \tag{1}$$

A language L is in Σ_3^{p} if and only if there exist a polynomial p and a polynomial-time decidable relation $R \subseteq \Sigma^* \times \Sigma^* \times \Sigma^* \times \Sigma^*$ such that for all words $x \in \Sigma^*$ we have

$$x \in L \iff (\exists w \in \Sigma^{p(|x|)})(\forall y \in \Sigma^{p(|x|)})(\exists z \in \Sigma^{p(|x|)})[R(x, w, y, z)]. \tag{2}$$

2.4. Succinct representations of general graphs

By *circuit* we refer to combinatorial circuits containing input-, output-, negation-, and-, and or-gates. The fan-in of each gate is at most 2. Fan-out is not restricted. For a circuit C with n input gates and m output gates, we also use C to denote the function computed by the circuit C. This function, C, maps elements of Σ^n to Σ^m. For a circuit C, we use $\mathrm{code}(C)$ to denote a standard binary encoding of the circuit. The exact details of such a coding will not be important, but note that for n-input and m-output circuits C the coding will have length at least $n + m$.

We use circuits to define graphs succinctly. For positive integers n, given an $2n$-input, 1-output circuit C, we say that it *specifies the graph* G whose vertex set is $V = \Sigma^n$ and whose edge set is defined as follows: There is an edge from $x \in V$ to $y \in V$ if and only if $C(xy) = 1$. We say that C is a *succinct representation of* G. Note that a graph G has many succinct representations.

In the rest of this article, we will describe different (often quite complicated) graphs and tournaments that are highly regular and can be described succinctly by circuits that are easy to compute (for instance, in logarithmic space). In the present article, we do not explain in detail how these succinct representations can be obtained, but interested readers will find detailed constructions in the technical report version [6] of the present article.

We next formalize the radius, diameter, and k-kingship problems for succinctly specified graphs. Let k be a fixed positive integer.

$$\textsc{succinct-}k\textsc{-radius} = \{\mathrm{code}(C) \mid \text{the graph specified by } C \text{ has radius at most } k\}.$$

$$\textsc{succinct-}k\textsc{-diameter} = \{\mathrm{code}(C) \mid \text{the graph specified by } C \text{ has diameter at most } k\}.$$

$$\textsc{succinct-}k\textsc{-king} = \{\langle \mathrm{code}(C), x\rangle \mid x \text{ is a } k\text{-king in the graph specified by } C\}.$$

2.5. Tournament family specifiers

We can use circuits as introduced in the previous section to describe tournaments succinctly. However, there is also a different, more computationally uniform (and less flexible) way of specifying tournaments succinctly.

A *tournament family specifier* is a function $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ such that

(i) f is a polynomial-time computable function;
(ii) f is commutative, that is, for all $x, y \in \Sigma^*$ we have $f(x, y) = f(y, x)$; and
(iii) f is a selector, that is, for all $x, y \in \Sigma^*$ we have $f(x, y) \in \{x, y\}$.

We interpret this as specifying, in the following way, a family of tournaments, one per length. At each length n, the vertices in the length-n tournament specified by f will be the bitstrings in Σ^n. For each two distinct vertices among these, x and y, the edge between them will be $x \to y$ if $f(x, y) = y$ and it will be $y \to x$ if $f(x, y) = x$. There will be no self-loops. Since our function f always chooses one of its inputs and is commutative, this indeed yields a family of tournaments. We call the tournament just described *the length-n tournament induced by f*. One may note that the constraints in the definition of tournament family specifiers apply even between strings of different lengths, and yet this is never used in our proofs of results about tournament family specifiers since specifiers specify different, separate tournaments at each length. The constraint just ensures that every tournament family specifier is a (commutative) P-selector in the sense of P-selectivity theory, but is not needed otherwise.

For completeness, we recall from the introduction the definition of the k-Kings_f problem for tournament family specifiers f, namely k-$\mathrm{Kings}_f = \{x \in \Sigma^* \mid x \text{ is a } k\text{-king in the length-}|x| \text{ tournament specified by } f\}$. A language L is called a P-*k-king language*, if there exists a tournament family specifier f with k-$\mathrm{Kings}_f = L$. Our main interest in this article is to find out which languages are P-k-king languages. The following lemma provides a first example (we will need the construction from the lemma in later proofs). Note that $\Sigma^* - \{1, 11\} = \mathrm{pad}'_j(\Sigma^*)$ holds for all positive integers j.

Lemma 2.10. *For every $k \geq 2$, the language $\Sigma^* - \{1, 11\}$ is a* P-*k-king language.*

Proof. We must show that there is a tournament family specifier such that for all word lengths n, except for $n = 1$ and $n = 2$, all words in Σ^n are k-kings of the length-n tournament induced by f. Note that all vertices of a tournament of length $n \geq 2$ are k-kings if and only if the tournament has diameter 2.

For $n = 0$, the tournament family specifier is trivial and, indeed, the single word ϵ is a k-king of the tournament whose vertex set is Σ^0.

For $n = 1$, the tournament family specifier directs the arrow from 0 to 1. So 0 and only 0 is a k-king of the tournament $\Sigma^1 = \Sigma$.

For $n = 2$, the tournament family specifier specifies the following tournament on $\Sigma^2 = \{00, 01, 10, 11\}$: The three vertices 00, 01, and 10 form a cycle and there is an edge from each of them to 11. Then clearly the three vertices 00, 01, and 10 are k-kings of Σ^2, and 11 is not.

For $n = 3$ and all larger n, the tournament family specifier for the tournament Σ^n specifies a diameter-2 tournament on 2^n vertices. By Lemma 2.8, such a tournament exists (we have $2^n \geq 8$). The trickier part is specifying such a tournament in polynomial time. We give a rather detailed construction in the following to give a flavor of the typical line of arguments; for similar constructions in later proofs, please see the technical report version [6].

Let us introduce names for the words of Σ^n: Let σ_i with $i \in \{1, 2, 3, \ldots, 2^n\}$ denote the ith lexicographical word in Σ^n. The tournament of diameter 2 on these words will be the tournament $T_{2^n}^{\text{diam}=2}(\sigma_1, \ldots, \sigma_{2^n})$ constructed in Lemma 2.8. It remains to show how the edge relation of this tournament can be decided in polynomial time.

The general rule resulting from the construction is the following: Suppose we are given two input words σ_i and σ_j for which our machine M should decide whether there is an edge from σ_i to σ_j or the other way round. We assume $i < j$, otherwise we exchange the roles of σ_i and σ_j (and for $i = j$ nothing needs to be done). If i and j are both at most 6, then M can direct the edge according to the hardwired tournament $T_6^{\text{diam}=2}$. So, suppose $j > 6$. In this case we distinguish two cases, depending on whether j is even or odd. First, assume that j is odd. Then M outputs that there is an edge from σ_i to σ_j. Second, assume that j is even. If $i = j - 1$, then M outputs that there is an edge from σ_i to σ_j. If $i < j - 1$, then M outputs that there is an edge from σ_j to σ_i. Clearly, the computation of M takes only polynomial time. Furthermore, M specifies exactly the tournament $T_{2^n}^{\text{diam}=2}(\sigma_1, \ldots, \sigma_{2^n})$ as can be seen by comparing the construction of $T_{2^n}^{\text{diam}=2}$ in Lemma 2.8 and the above specification of M's output. □

3. Complexity of the diameter problem

In this section, we prove that the succinct diameter problem is complete for Π_2^{p}. Indeed, we show that the succinct diameter problem restricted to tournaments, defined by

$$\text{SUCCINCT-}k\text{-DIAMETER-TOURNAMENT} = \{\text{code}(C) \mid \text{the graph specified by } C \text{ is a tournament of diameter at most } k\},$$

is already hard for this class for every fixed $k \geq 2$. The tools that we introduce for the proof of this result will be important in the following sections.

Definition 3.1. An *ℓ-layered tournament* is a tournament whose vertex set is the disjoint union of ℓ nonempty sets $L_1, \ldots, L_\ell$ such that the following holds: For any vertex $u \in L_i$ and $v \in L_j$ with $i < j - 1$, there is an edge $v \to u$.

Lemma 3.2. *Let T be an ℓ-layered tournament, let $u \in L_1$, and let $v \in L_\ell$. Then the shortest path from u to v has length at least $\ell - 1$.*

Proof. Each edge of a path from u to v can increase the level by at most 1. □

Our next task is the definition of a rather complex tournament that will be used in later proofs. Recall that we want to show that every problem in Π_2^{p} reduces to the succinct k-diameter problem. For this, we construct a tournament in which all vertices are k-kings, except possibly for one vertex, which will be a k-king exactly if a certain "for all … exists …" property is true. The tournament is visualized in Fig. 1.

Definition 3.3. Let $n \geq 3$ and $k \geq 2$ be integers and let $R \subseteq \Sigma^n \times \Sigma^n$ be a relation. Let V be a vertex set. Let $J \subseteq V$ be a set of even cardinality (its members will be called the *junk vertices*) such that $|V| - |J| = k + 8 + 2 \cdot 2^n$ for even k and $|V| - |J| = k + 7 + 2 \cdot 2^n$ for odd k. We define a tournament $T^k(R, J)$ on the vertex set V as follows.

(i) *The layers:* The tournament is a $(k+1)$-layered tournament. The layers get the following special names: Layer L_1 is the *potential k-king layer*. Layers $L_2, \ldots, L_{k-1}$ are the *antenna layers* (note that for $k = 2$ there are no antenna layers). Layer L_k is the *z-layer*. Layer L_{k+1} is the *y-layer*.

(ii) *Vertices in the layers:* The potential k-king layer L_1 contains a single vertex p. Each antenna layer L_i for $i \in \{2, \ldots, k-1\}$ also contains a single vertex a_i. The z-layer L_k is the set $Z \cup Z'$, where $Z' = \{z_1, z_2, z_3\}$ for even k and $Z' = \{z_1, z_2\}$ for odd k. The set $Z = \{\beta_z \mid z \in \Sigma^n\}$ contains 2^n elements that are indexed by the bitstrings of length n. The y-layer L_{k+1} is the set $Y \cup C \cup D \cup J$, where $Y = \{\alpha_y \mid y \in \Sigma^n\}$ also contains 2^n elements that are indexed by the bitstrings of length n and $C = \{c_0, c_1, c_2\}$ and $D = \{d_0, d_1, d_2\}$. We assume that all the sets Y, C, D, Z, Z', and J are pairwise disjoint.

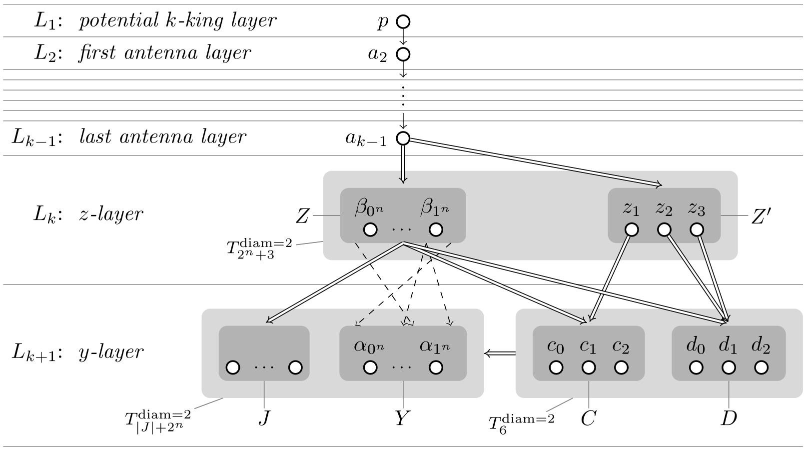

Fig. 1. Visualization of the tournament $T^k(R, J)$ for even k; for odd k the vertex z_3 is not present. Vertices are depicted as small circles. Wherever there is no arrow between two vertices in different layers, the missing arrow points upward. The darker areas denote sets of vertices, a double-line arrow from an area (or a single vertex) to another area means that there is an arrow from every vertex in the first area to every vertex in the second area. The larger lighter areas denote sets of vertices that are internally connected so that they form the tournament $T_x^{\mathrm{diam}=2}$. The dashed arrows between Z and Y indicate examples of how the arrows between these two sets might be directed; they — and only they — depend on R.

(iii) *Edges between adjacent layers:* We now describe the connections between one layer and the next (the connections between vertices in nonadjacent layers are already fixed by the fact that $T^k(R, J)$ is a layered tournament). For layers $L_1, \ldots, L_{k-1}$, each of which contains only one vertex, there is an edge to every vertex in the next layer. For layers L_k and L_{k+1}, we explain for each element of $Y \cup C \cup D \cup J$ of vertices in the y-layer which edges there are to or from vertices in the z-layer.
  (a) Let $\alpha_y \in Y$. For every vertex $\beta_z \in Z$, there is (a) an edge $\beta_z \to \alpha_y$ if and only if $(y, z) \in R$ and (b) an edge $\alpha_y \to \beta_z$ if and only if $(y, z) \notin R$. For every vertex $z_i \in Z'$, there is an edge $\alpha_y \to z_i$.
  (b) Let $c \in C$. For every $z \in Z$, there is an edge $z \to c$. There is also an edge $z_1 \to c$. However, there is an edge $c \to z_2$ and, provided z_3 exists, also an edge $c \to z_3$.
  (c) Let $d \in D$. For every $z \in Z$, there is also an edge $z \to d$. However, there is an edge $d \to z_1$; and there is an edge $z_2 \to d$ and, provided z_3 exists, also an edge $z_3 \to d$.
  (d) Let $j \in J$. For every $z \in Z$ there is an edge $z \to j$. However, for every $z_i \in Z'$ there is an edge $j \to z_i$.

(iv) *Edges inside each layer:* For the potential k-king layer and the antenna layers, nothing needs to be specified. For the z-layer, we connect the vertices inside the z-layer such that they form the tournament $T^{\mathrm{diam}=2}_{2^n+2}$ or the tournament $T^{\mathrm{diam}=2}_{2^n+3}$ from Definition 2.9. For the y-layer, we connect the vertices as follows.
  (a) Connect the vertices in $Y \cup J$ such that they form the tournament $T^{\mathrm{diam}=2}_{2^n+|J|}$ from Lemma 2.8 for the given numbering.
  (b) Connect the vertices in $C \cup D$ such that they form the tournament $T^{\mathrm{diam}=2}_{6}$ from Definition 2.6.
  (c) Connect every vertex $u \in Y \cup J$ and every vertex $v \in C \cup D$ by an edge $v \to u$.

Note that in the above definition of $T^k(R, J)$ we frequently referred to n, where $R \subseteq \Sigma^n \times \Sigma^n$. If R happens to be the empty relation, then n is not specified uniquely, and one would have to write something like $T^k(R, J, n)$ to be precise. To keep the notation simple, we write only $T^k(R, J)$.

We will often need to talk about "an arbitrary vertex in some layer L_i." We will generally use the variables l_i, l_i' and so on to denote such vertices.

Lemma 3.4. *Let $k \geq 2$, let $n \geq 3$, let $R \subseteq \Sigma^n \times \Sigma^n$ be a relation, and let J be a set of even size. Then the tournament $T^k(R, J)$ has the following properties:*

(i) *The vertex p in the potential k-king layer is a k-king if and only if for every $y \in \Sigma^n$ there exists a $z \in \Sigma^n$ such that $(y, z) \in R$ holds.*
(ii) *All other vertices are k-kings of the tournament.*

Proof. We start with the proof of the first claim.

For the first direction, assume that for every $y \in \Sigma^n$ there exists a $z \in \Sigma^n$ such that $(y, z) \in R$ holds. Then p is a k-king as can be seen as follows: The construction states that, except for layers L_k and L_{k+1}, we can go from every layer in one step to all vertices in the next layer. Thus, we can reach all vertices in the z-layer L_k within $k - 1$ steps. To reach the vertices in the y-layer from the z-layer in another step, we use the following edges:

(i) Let $\alpha_y \in Y$ with $y \in \Sigma^n$. By assumption, there exists a $z \in \Sigma^n$ with $(y, z) \in R$. By part a of Definition 3.3.iii there is an edge $\beta_z \to \alpha_y$.
(ii) Let $y \in C \cup D \cup J$. Then $\beta_z \to y$ for any $\beta_z \in Z$ by part b, c, or d.

For the second direction, assume that p is a k-king. Let $y \in \Sigma^n$ be given. We have to argue that there exists a $z \in \Sigma^n$ with $(y, z) \in R$. Consider a path from p to α_y of length at most k. By Lemma 3.2, this path has to end with a step along an edge from some vertex $l_k \in L_k$ to α_y. This implies $l_k \in Z$ since there is no edge from any vertex in $Z' = L_k - Z$ to α_y, see part a of Definition 3.3.iii. This shows $l_k = \beta_z$ for some $z \in \Sigma^n$ and the presence of the edge $\beta_z \to \alpha_y$ implies $(y, z) \in R$ as claimed, again by part a.

We next prove the second claim by going over all other vertices of $T^k(R, J)$ one by one and showing that they are k-kings.

(i) Consider the vertex a_i in an antenna layer L_i. From a_i we can reach all vertices in the z-layer within $k - 2$ steps and all vertices in the antenna layers L_j with $i < j$ in even fewer steps. From the z-layer, we can reach all vertices in layers L_j with $j < i$ in one step.

Consider a vertex l_{k+1} in the y-layer. We show that we can reach it within two steps from some vertex in the z-layer. For $l_{k+1} \in Y \cup J$, we can use the path $z_1 \to c \to l_{k+1}$, where $c \in C$ is an arbitrary vertex. For $l_{k+1} \in C$, we can use the path $z_1 \to l_{k+1}$. Finally, for $l_{k+1} \in D$, we can use the path $z_2 \to l_{k+1}$.

(ii) Consider a vertex l_k in the z-layer L_k. We can reach every other vertex of the z-layer from l_k within two steps since the vertices of the z-layer induce the tournament $T^{\mathrm{diam}=2}_{2^n+2}$ or $T^{\mathrm{diam}=2}_{2^n+3}$. Next, by Definition 3.3.iii parts b and c there always exists a vertex $y \in C \cup D$ such that $l_k \to y$. This implies that we can reach the potential k-king and all vertices in the antenna layers through the paths $l_k \to y \to p$ and $l_k \to y \to a_i$. It remains to argue that we can reach all vertices l_{k+1} in the y-layer within k steps.

If $l_k \in Z$, we can reach l_{k+1} in (a) one step for $l_k \in C \cup D \cup J$ and (b) in two steps for $l_{k+1} \in Y$, namely through the path $l_k \to c \to l_{k+1}$, where $c \in C$ is arbitrary.

If $l_k = z_1$, we can reach l_{k+1} as follows:

(a) Let $l_{k+1} \in Y \cup J$. Then $l_k \to c_1 \to l_{k+1}$ is the desired path.
(b) Let $l_{k+1} \in C$. Then $l_k \to l_{k+1}$ is the desired path.
(c) Let $l_{k+1} \in D$. By Lemma 2.7 part iii, there exists a $c \in C$ such that $c \to l_{k+1}$. Then $l_k \to c \to l_{k+1}$ is the desired path.

If $l_k \in \{z_2, z_3\}$, we can reach the vertices of the y-layer as follows:

(a) Let $l_{k+1} \in Y \cup J$. Then $l_k \to d_1 \to l_{k+1}$ is the desired path.
(b) Let $l_{k+1} \in C$. By Lemma 2.7 part ii, there exists a $d \in D$ such that $d \to l_{k+1}$. Then $l_k \to d \to l_{k+1}$ is the desired path.
(c) Let $l_{k+1} \in D$. Then $l_k \to l_{k+1}$ is the desired path.

(iii) Consider a vertex l_{k+1} in the y-layer L_{k+1}. Through p, we can reach all vertices in layers L_1 to L_k within k steps. Now, consider the vertices in the y-layer.

(a) Let $l_{k+1} \in Y \cup J$. Since the vertices in $Y \cup J$ form a subtournament of $T^k(R, J)$ of diameter 2, there is a path of length at most 2 from l_{k+1} to every $l'_{k+1} \in Y \cup J$ inside this subtournament. Next, consider $l'_{k+1} \in C$. This can be reached from l_{k+1} through the path $l_{k+1} \to z_1 \to l'_{k+1}$, see parts a and d of Definition 3.3.iii for $l_{k+1} \to z_1$ and part b for $z_1 \to l'_{k+1}$. Finally, consider $l'_{k+1} \in D$. Then $l_{k+1} \to z_2 \to l'_{k+1}$ is the desired path.
(b) Let $l_{k+1} \in C \cup D$. Then all $l'_{k+1} \in Y \cup J$ can be reached in one step. All $l'_{k+1} \in C \cup D$ can be reached within two steps, since the vertices in $C \cup D$ form a subtournament of T of diameter 2. $\square$

In order to represent a tournament $T^k(R, J)$ succinctly using a circuit, the vertex set of the tournament must be Σ^m for a sufficiently large m. The following definition introduces a special notation for the resulting tournament. As mentioned earlier, we omit the details of which words are used, exactly, to encode the antenna vertices, Y, Z, J, C, and D, and we also omit a description of the circuit that specifies the tournament. In both cases, interested readers will find the detailed constructions in the technical report version [6].

Definition 3.5. Let $k \geq 2$ and $n \geq 3$ be integers, let $R \subseteq \Sigma^n \times \Sigma^n$, and let m be an integer such that $2^m > k + 8 + 2 \cdot 2^n$. Then, $T^k_{\Sigma^m}(R)$ denotes the tournament $T^k(R, J)$, where the vertex set V is Σ^m, where the potential k-king is encoded as 0^m, and where the set $J \subseteq V$ of junk vertices has size $2^m - k - 8 - 2 \cdot 2^n$ for even k and size $2^m - k - 7 - 2 \cdot 2^n$ vertices for odd k.

Theorem 3.6. *Let $k \geq 2$. Then* SUCCINCT-k-DIAMETER-TOURNAMENT *is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for Π^{p}_2.*

Proof. The problem SUCCINCT-k-DIAMETER-TOURNAMENT is clearly a member of Π^{p}_2: Given as input a circuit X, we must check (a) whether the graph specified by X is a tournament and (b) whether in this graph for all ordered pairs (s, t) of vertices there exists a path of length at most k from s to t, that is, whether there exist a list of $k - 1$ (or $k - 2$ or … or 0) appropriate intermediate vertices. The first test is a coNP-test, the second is a $\mathrm{coNP}^{\mathrm{NP}}$-test.

To prove hardness, let an arbitrary language $L \in \Pi^{\mathrm{p}}_2$ be given. Then there exist a polynomial p and a ternary relation $R \subseteq \Sigma^* \times \Sigma^* \times \Sigma^*$ such that Eq. (1) from Section 2.3 holds. We describe a polynomial-time reduction from L to SUCCINCT-k-DIAMETER-TOURNAMENT. Let an input x be given. We define a relation $R_x \subseteq \Sigma^{p(|x|)} \times \Sigma^{p(|x|)}$ as follows: Let $(y, z) \in R_x$ for $y, z \in \Sigma^{p(|x|)}$ if and only if $(x, y, z) \in R$. Compute the minimal m such that $2^m > k + 8 + 2 \cdot 2^{p(|x|)}$. We map x to a circuit X that specifies the tournament $T^k_{\Sigma^m}(R_x)$. By Lemma 3.4, all vertices of this tournament are k-kings of T, except possibly for $p = \sigma_1$. This vertex is a k-king of T if and only if for all $y \in \Sigma^{p(|x|)}$ there is a $z \in \Sigma^{p(|x|)}$ such that $(y, z) \in R_x$. But this means that p is a k-king if and only if $x \in L$. Thus, $x \in L$ if and only if *all* vertices of $T^k_{\Sigma^m}(R_x)$ are k-kings. $\square$

The proof of Theorem 3.6 also proves the following corollary.

Corollary 3.7. *Let $k \geq 2$. Then,* SUCCINCT-k-DIAMETER *is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for Π^{p}_2.*

Note that the language SUCCINCT-1-DIAMETER is easily seen to be $\leq_m^p$-complete for coNP while the language SUCCINCT-1-TOURNAMENT-DIAMETER is the empty set (since in our model all graphs specified by a circuit have size at least 2, and so in the case of tournaments, cannot have diameter 1, since no tournament on 2 or more vertices has diameter 1).

4. Complexity of P-k-king languages

In this section, we establish the following result, which is the central structural result of this article.

Theorem 4.1. *Let $k \geq 2$. Each language in $\Pi_2^p - \{\emptyset, \Sigma^*\}$ is $\leq_m^p$-equivalent to a* P-*k-king language.*

This result says that, excluding from our attention the singleton degrees of the empty set and of Σ^*, for every $k \geq 2$ every many-one degree can be named by a k-kings problem. That is, k-kings problems are so flexible that they take on every possible Π_2^p complexity level.[1]

Note in particular that the theorem applies to the complete Π_2^p degree. Thus, we have the following corollary, which shows that the result of [8] that P-2-king languages are all in Π_2^p is optimal.

Corollary 4.2. *For each $k \geq 2$, there is a Π_2^p-complete* P-*k-king language.*

We will prove Theorem 4.1 by showing a result, Theorem 4.3, that in effect is even stronger.

Theorem 4.3. *Let L be a language and let $k \geq 2$. Then $L \in \Pi_2^p$ if and only if there exists a positive integer j such that $\mathrm{pad}_j'(L)$ is a* P-*k-king language.*

Theorem 4.3 says that each set in Π_2^p has a padded version of itself that is a P-k-king language. By part v of Lemma 2.2, this implies Theorem 4.1. Note that Theorem 4.3, unlike Theorem 4.1, holds even for the trivial languages $\emptyset$ and Σ^*.

One might naturally ask why it is worth proving Theorem 4.3 rather than merely proving Theorem 4.1. After all, proving Theorem 4.1 directly is slightly simpler; pad_j' is not the most intuitively natural padding family in the world; and there are costs to our approach such as the fact that the proof of Theorem 4.3 has much machinery and is somewhat fragile. However, it turns out that Theorem 4.3 is "what we really need" in certain applications: Our proof that the radius problem for succinctly specified graphs is Σ_3^p-complete relies on the stronger claim of Theorem 4.3. Theorem 4.3 also is worthwhile simply for what it says: It shows that a very simple, restricted family of reductions—namely, those of the family pad_j'—suffices to map each set in $\Pi_2^p - \{\emptyset, \Sigma^*\}$ to a P-k-king language it is equivalent to.

To prove Theorem 4.3, we introduce a new tournament that "weaves together" multiple $T_{\Sigma^m}^k(R)$ tournaments. The woven tournament is visualized in Fig. 2.

Definition 4.4. Let $k \geq 2$, $n \geq 1$, and $n' \geq 3$ be integers, let $R \subseteq \Sigma^n \times \Sigma^{n'} \times \Sigma^{n'}$ be a ternary relation, let m be an integer with $2^m > k + 8 + 2 \cdot 2^{n'}$, and let F (called the *fill-up vertices*) be a set of vertices whose cardinality is not 0, 2, or 4. The *woven tournament* $W^k(R, F, m)$ is defined as follows.

(i) *The tournaments that are woven together:* The vertex set of the woven tournament is the disjoint union of all $T_{\Sigma^m}^k(R_x)$ for $x \in \Sigma^n$ and of F. Here, $R_x \subseteq \Sigma^{n'} \times \Sigma^{n'}$ is the relation that contains all pairs (y, z) such that $(x, y, z) \in R$. Taking the disjoint union means that for every $x \in \Sigma^n$ and each $v \in T_{\Sigma^m}^k(R_x)$, the tournament $W^k(R, F, m)$ contains a tagged element v^x. Tagged potential k-kings get a name for later reference: Let the tagged version of the single vertex of the potential k-king layer of $T_{\Sigma^m}^k(R_x)$ be denoted p^x in the following.

(ii) *Edges inside the tournaments that are woven together:* The vertices inside the woven tournament that come from one $T_{\Sigma^m}^k(R_x)$ are connected in the woven tournament in the same way as they are connected in $T_{\Sigma^m}^k(R_x)$; formally, there is an edge $u \to v$ in $T_{\Sigma^m}^k(R_x)$ if and only if there is an edge $u^x \to v^x$ in $W^k(R, F, m)$.

(iii) *The layers:* The woven tournament is a $k+1$-layered tournament. The vertices on layer L_i of the woven tournament are exactly the vertices in the tournaments $T_{\Sigma^m}^k(R_x)$ in layer L_i. In addition, all vertices from F are part of layer L_k.

(iv) *Edges between different, nonadjacent layers:* The definition of a layered tournament, Definition 3.1, fixes how vertices in nonadjacent layers are connected.

(v) *Edges inside each layer:* For two vertices in the same $T_{\Sigma^m}^k(R_x)$, we have already fixed how they are connected. In the other cases, we proceed as follows:
   (a) For vertices in layers other than layer L_k, we add the missing edges arbitrarily.
   (b) For vertices in layer L_k, we can also connect vertices in different $T_{\Sigma^m}^k(R_x)$ arbitrarily, but for a vertex $u \in T_{\Sigma^m}^k(R_x)$ and a vertex $f \in F$, the edge direction is $f \to u^x$.
   (c) Let $f_1, \ldots, f_{|F|}$ be names for the vertices inside F. We connect them such that they form the tournament $T_{|F|}^{\mathrm{diam}=2}(f_1, \ldots, f_{|F|})$.

1 The case of the degree of the empty set is hopeless, since the only set $\leq_m^p$-equivalent to the empty set is the empty set, and the empty set can never be a king language. The case of the degree of Σ^* depends on such things as the alphabet cardinality. However, note that when $\Sigma = \{0, 1\}$, as it does throughout this article, the two length-1 strings already ensure that no king language can be Σ^*, since at least one of those two strings is not a king in the length-1 tournament. So, since the only set $\leq_m^p$-equivalent to Σ^* is Σ^*, this means that the degree of Σ^* cannot contain a king language.

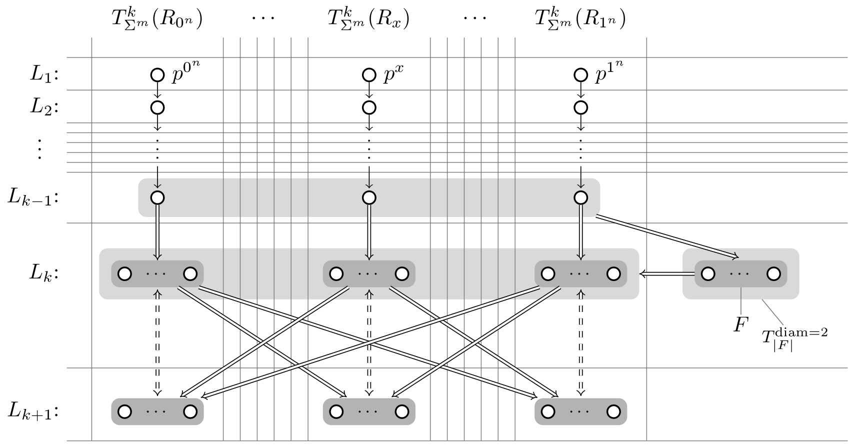

Fig. 2. Visualization of the woven tournament $W^k(R, F, m)$. It consists of the different $T^k_{\Sigma^m}(R_x)$ for $x \in \Sigma^n$, shown as vertical columns, plus the set F, which lies in layer L_k. Inside each column, the vertices are connected exactly as in $T^k_{\Sigma^m}(R_x)$; the dashed line between the vertices on the last two layers indicates that the directions of the edges vary there (they depend on R_x). As in Fig. 1, missing edges between vertices on different layers point upward and double-line arrows indicate the direction of all edges between two sets. Missing edges inside a layer can be directed arbitrarily, except — as indicated — inside F.

(vi) *Edges between adjacent layers:* Once more, the connection between vertices u^x and v^x stemming from the same $T^k_{\Sigma^m}(R_x)$ is already fixed. For all vertex pairs in adjacent layers for which we have not yet assigned a direction, we use the following rules:
   (a) The edges between the vertices in layer L_{k-1} and the vertices in F always point to the vertex in F.
   (b) Let l^x_k be a vertex in layer L_k and let $l^{x'}_{k+1}$ be a vertex in layer L_{k+1}, where $x \neq x'$. Then the edge between l^x_k and $l^{x'}_{k+1}$ is $l^x_k \to l^{x'}_{k+1}$.
   (c) Except when otherwise specified by the two just-stated rules, edges point from the vertex in the layer with the larger index to the vertex in the layer with the smaller index.

Similarly to the definition of $T^k(R, J)$, the definition of $W^k(R, F, m)$ makes references to the numbers n and n', which are not unique if R happens to be the empty relation. To keep the notation simple, we write $W^k(R, F, m)$ nevertheless.

Lemma 4.5. *Let $k \geq 2$, $n \geq 1$, and $n' \geq 3$ be integers, let $R \subseteq \Sigma^n \times \Sigma^{n'} \times \Sigma^{n'}$ be a ternary relation, let m be an integer with $2^m > k + 8 + 2 \cdot 2^{n'}$, and let F be a set of vertices whose cardinality is not 0, 2, or 4. Then the woven tournament $W^k(R, F, m)$ has the following properties:*

(i) *For every $x \in \Sigma^n$ the vertex p^x is a k-king of the woven tournament if and only if for every $y \in \Sigma^{n'}$ there exists a $z \in \Sigma^{n'}$ such that $(x, y, z) \in R$ holds.*
(ii) *All other vertices (vertices other than the p^x) are k-kings of the woven tournament.*

Proof. We start with a proof of the first claim.

For the first direction, assume that p^x is a k-king of the woven tournament. We have to argue that for every $y \in \Sigma^{n'}$ there exists a $z \in \Sigma^{n'}$ such that $(x, y, z) \in R$. Consider the tournament $T^k_{\Sigma^m}(R_x)$. Let V be its vertex set and let $V^x = \{v^x \mid v \in V\}$ be the set of tagged versions of the vertices in V. Then V^x is a set of vertices in the woven tournament.

We claim that every path of length at most k from p^x to a vertex $v^x \in V^x$ in the y-layer has to fall completely inside V^x. Since the woven tournament is a layered tournament, by Lemma 3.2 we know that a path from p^x to v^x has to advance one layer in each step. In particular, this path cannot go back a level or stay inside the same level for one step.

Suppose the path from p^x to v^x leaves the set V^x at some point (and returns to it later on, at the latest when it reaches $v^x \in V^x$). When the path leaves V^x, it must do so while advancing a level. By the construction of the woven tournament, this is possible only in the following cases: First, we can go from a vertex in layer L_{k-1} inside V^x to any vertex in F. However, there is no edge from any vertex in F to any vertex in the next layer, so we cannot advance from F to the last layer. Second, we can go from any $l^x_k \in V^x$ in layer L_k directly to every $l^{x'}_{k+1}$ in layer L_{k+1}, but only for $x' \neq x$ and not, as one would need, for $x = x'$.

We now know that the path from p^x to v^x must fall within V^x. This shows that there is a path of length at most k from the potential k-king of $T^k_{\Sigma^m}(R_x)$ to every vertex in the y-layer of $T^k_{\Sigma^m}(R_x)$. This implies that the potential k-king of $T^k_{\Sigma^m}(R_x)$ is a k-king of $T^k_{\Sigma^m}(R_x)$, since all vertices in layers other than the y-layer are easily reachable even within $k - 1$ steps from the potential k-king. By the first part of Lemma 3.4, this implies that for every $y \in \Sigma^{n'}$ there exists a $z \in \Sigma^{n'}$ such that $(y, z) \in R_x$, which is equivalent to $(x, y, z) \in R$.

Let us now prove the second direction of the first claim. Assume that for every $y \in \Sigma^{n'}$ there exists a $z \in \Sigma^{n'}$ such that $(x, y, z) \in R$ holds and we wish to show that p^x is a k-king of the woven tournament.

By Lemma 3.4, the potential k-king p of $T^k_{\Sigma^m}(R_x)$ is a k-king of $T^k_{\Sigma^m}(R_x)$. This implies that there is a path in the woven tournament of length at most k from p^x to each vertex v^x for $v \in V^x$. Again, V^x is the tagged version of the vertex set of $T^k_{\Sigma^m}(R_x)$.

The harder part is arguing that we can reach all vertices of the woven tournament *other* than those in V^x from p^x within k steps. Note that this is *always* the case, independently of whether $(\forall y)(\exists z)[R(x, y, z)]$ holds or not:

(i) Consider any vertex $l_i^{x'}$ for $x' \neq x$ in any layer L_i for $i \leq k-1$. This vertex can be reached as follows: There is a path of length $k-1$ from p^x to every vertex in V^x in layer L_k, and there is an edge from every such vertex to $l_i^{x'}$.
(ii) Consider any vertex in F. This vertex can be reached as follows: There is a path of length $k-2$ from p^x to the single vertex of V^x in layer L_{k-1}, and there is an edge from this vertex to every vertex in F. Thus, we can reach every vertex of F within $k-1$ steps from p^x.
(iii) Consider any vertex $l_k^{x'}$ for $x' \neq x$ in layer L_k. This vertex can be reached from p^x through the path of length $k-1$ to some vertex $f \in F$ (note that F contains at least one vertex) and the edge from f to $l_k^{x'}$.
(iv) Consider any vertex $l_{k+1}^{x'}$ for $x' \neq x$ in layer L_{k+1}. This vertex can be reached as follows: There is a path of length $k-1$ from p^x to each vertex $l_k^x \in V^x$ in layer L_k and, since $x' \neq x$, there is an edge from l_k^x to $l_{k+1}^{x'}$.

We next prove the second claim. For this claim, we show that all vertices that are not in the potential k-king layer are always k-kings of the woven tournament.

(i) Let a_i^x be a vertex in an antenna layer L_i. From a_i^x, we can reach all $u^x \in V^x$ within k steps by Lemma 3.4 and the fact that the woven tournament inherits the edges inside V^x from $T^k_{\Sigma^m}(R_x)$. Now, consider a vertex v of the woven tournament outside V^x. Above, we argued that we can reach v from p^x within k steps. Furthermore, the path from p^x to v goes through a_i^x. Thus, we can also reach v from a_i^x within k steps.
(ii) Let l_k^x be a vertex in layer L_k (but not a vertex from F).
 (a) Again, we can reach all other vertices in V^x within k steps by Lemma 3.4.
 (b) Consider a vertex $l_i \notin V^x$ in a layer L_i for $i < k$. Then, there is a direct edge $l_k^x \to l_i$.
 (c) Consider a vertex $l_k^{x'}$ for $x' \neq x$ in layer L_k. It can be reached within k steps from l_k^x as follows: There is an edge form l_k^x to $p^{x'}$ and there is a path of length at most $k-1$ from $p^{x'}$ to $l_k^{x'}$.
 (d) Consider a vertex $f \in F$. It can be reached within k steps from l_k^x by investing one step to go to any $p^{x'}$ with $x' \neq x$ and $k-1$ steps to reach f.
 (e) Every vertex $l_{k+1}^{x'}$ for $x' \neq x$ in layer L_{k+1} can be reached in one step.
(iii) Let $f \in F$. From this vertex, we can reach every vertex l_k^x in layer L_k in one step. We can reach every vertex l_i^x in any different layer in two steps as follows: Use one step to go to some $(l'_k)^{x'}$ in layer L_k with $x' \neq x$ and use another step to go to l_i^x. Note that this works both for the antenna layers and for the y-layer. Finally, we can reach all other vertices in F within two steps, since the vertices in F are connected according to the edge relation of $T^{\text{diam}=2}_{|F|}$.
(iv) Consider a vertex l_{k+1}^x in layer L_{k+1}.
 (a) As before, we can reach all vertices $v^x \in V^x$ within k steps by Lemma 3.4.
 (b) We can reach all vertices $l_i^{x'}$ with $x' \neq x$ in any layer L_i for $i \leq k$ as follows: There is an edge $l_{k+1}^x \to p^{x'}$ and a path of length at most $k-1$ from $p^{x'}$ to $l_i^{x'}$.
 (c) All $f \in F$ can be reached in one step from l_{k+1}^x.
 (d) Let $(l'_{k+1})^{x'}$ for $x' \neq x$ be a vertex in layer L_{k+1}. To reach it, we invest one step to reach either the vertex z_1^x or the vertex z_2^x. By Definition 3.3 part iii, at least one such edge always exists. Then we can reach $(l'_{k+1})^{x'}$ from either z_1^x or z_2^x in one step. □

Our next step is to fix how the vertices of the tournament $W^k(R, F, m)$ are coded.

Definition 4.6. Let $k \geq 2$, $n \geq 0$, and $n' \geq 3$ be integers, let $R \subseteq \Sigma^n \times \Sigma^{n'} \times \Sigma^{n'}$ be a ternary relation, let m be an integer such that $2^m > k + 8 + 2 \cdot 2^{n'}$, and let $l = n + m + 3$. The tournament $W^k_{\Sigma^l}(R)$ is defined as follows. For $n \geq 1$, its vertex set is $V = \Sigma^l$. The set F is the set of all elements of V that do *not* end with 000. Note that this set does not have size 0, 2, or 4. We number the vertices in F lexicographically. The vertices of the different $T^k_{\Sigma^m}(R_x)$ are encoded as follows: A vertex u^x is mapped to the bitstring $xu000$. Thus, we prefix the vertices of $T^k_{\Sigma^m}(R_x)$ with x and add 000 at the end. For $n = 0$, we also define a tournament $W^k_{\Sigma^l}(R)$, but differently (because $W^k(R, F)$ is not defined for $n = 0$). We set $W^k_{\Sigma^l}(R, m)$ to be the tournament $T^k_{\Sigma^{m+3}}(R_\epsilon)$.

We are now ready to prove Theorem 4.3.

Proof of Theorem 4.3. We start with the easier direction. Suppose $\text{pad}'_j(L)$ is a P-k-king language. Every P-k-king language is in Π_2^{p} as we have only to check on input x whether for all y of the same length there exist an integer $k' \in \{0, \dots, k\}$ and bitstrings $w_0, w_1, \ldots, w_{k'}$ of length $|x|$ such that $x = w_0$, $y = w_{k'}$, and for each $\ell \in \{0, \ldots, k'-1\}$, the selector on input $(w_\ell, w_{\ell+1})$ picks $w_{\ell+1}$. Next, if $\mathrm{pad}'_j(L) \in \Pi_2^{\mathrm{p}}$, then we clearly also have $L \in \Pi_2^{\mathrm{p}}$.

For the other direction, let $L \in \Pi_2^{\mathrm{p}}$ be given. By the quantifier characterization of the polynomial hierarchy, see Eq. (1) on Section 2.3, there exist a polynomial p and a polynomial-time decidable relation $R \subseteq \Sigma^* \times \Sigma^* \times \Sigma^*$ such that for all words $x \in \Sigma^*$ we have

$$x \in L \iff (\forall y \in \Sigma^{p(|x|)})(\exists z \in \Sigma^{p(|x|)})[R(x, y, z)].$$

Choose $j \geq 3$ such that the function $m(n) = n^j + j$ has the property that $2^{m(n)} > k + 8 + 2 \cdot 2^{p(n)}$. We claim that $\mathrm{pad}'_j(L)$ is a P-k-king language.

For each nonnegative integer n, we define a relation S_n as follows: $S_n \subseteq \Sigma^n \times \Sigma^{p(n)} \times \Sigma^{p(n)}$ and a tuple $(x, y, z) \in \Sigma^n \times \Sigma^{p(n)} \times \Sigma^{p(n)}$ is in S_n if and only if $(x, y, z) \in R$.

We must now argue that there is a tournament family specifier whose k-king sets for the different word levels are exactly the words in $\mathrm{pad}'_j(L)$. Observe that $\mathrm{pad}'_j(L) \subseteq \mathrm{pad}'_j(\Sigma^*) = \Sigma^* - \{1, 11\}$ and that $\mathrm{pad}'_j(L)$ differs from $\mathrm{pad}'_j(\Sigma^*)$ only for words of lengths l of the form $l = n + n^j + j + 3$ for some n.

We first explain which tournaments T_l for $l \geq 0$ we are going to use for each word length l. We distinguish two different kinds of word lengths l.

(i) For a word length l that is not of the form $l = n + n^j + j + 3$, let T_l be the tournament specified for the word length l by the selector f' constructed in the proof of Lemma 2.10. In that lemma, it is shown that $\mathrm{pad}'_j(\Sigma^*)$ is a P-k-king language.

(ii) For a word length l of the form $l = n + n^j + j + 3$ for some n let T_l be the tournament $W^k_{\Sigma^l}(S_n)$. Note that, indeed, the vertex set of this tournament is $\Sigma^l = \Sigma^{n+m(n)+3}$.

We claim that the k-king sets of the T_l form the language $\mathrm{pad}'_j(L)$.

(i) For word lengths l that are not of the form $n + n^j + j + 3$ we have $\mathrm{pad}'_j(L) \cap \Sigma^l = \mathrm{pad}'_j(\Sigma^*) \cap \Sigma^l$ and the k-king set of the tournament T_l is exactly $\mathrm{pad}'_j(\Sigma^*) \cap \Sigma^l$.

(ii) For word lengths l of the form $n + n^j + j + 3$ we have $T_l = W^k_{\Sigma^l}(S_n)$. The k-king set of T_l contains all bitstrings, except possibly for the p^x, which have the form $x0^{n-l}$. A vertex p^x is a k-king if and only if $x \in L$. This, in turn, is the case if and only if $\mathrm{pad}_j(x) \in \mathrm{pad}'_j(L)$. This shows that the k-king set of T_l is exactly $\mathrm{pad}'_j(L) \cap \Sigma^l$.

Clearly, the edge relations of the T_l can be decided in polynomial time. □

5. Complexity of the radius problem

In this section, we apply our results on P-k-king languages to prove that the succinct radius problem for directed graphs is complete for Σ_3^{p}. Note that in the present section, unlike in the rest of this article, we consider arbitrary directed graphs instead of just tournaments. The reason for this is simple. It is well known that every tournament has a 2-king [10]. So the k-radius problem, $k \geq 2$, is trivial for tournaments, since their radius is *always* at most 2.

Theorem 5.1. *Let $k \geq 2$. Then* SUCCINCT-k-RADIUS *is $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete for Σ_3^{p}.*

Proof. First, SUCCINCT-k-RADIUS $\in \Sigma_3^{\mathrm{p}}$ as can be seen as follows: Given as input the code of a $2n$-input circuit C we have to check whether *there exists* a length-n bitstring x such that *for all* length-n bitstrings $y \neq x$ *there exists* an integer $\ell \in \{0, \ldots, k-1\}$ and *there exist* ℓ bitstrings $z_1, \ldots, z_\ell$ of length n such that in the graph specified by C the following is a path: $x \to z_1 \to \cdots \to z_\ell \to y$ (for $\ell = 0$ the path will simply be $x \to y$, of course).

Second, let any language $L \in \Sigma_3^{\mathrm{p}}$ be given. Then, there exists a quaternary relation $R \subseteq \Sigma^* \times \Sigma^* \times \Sigma^* \times \Sigma^*$ and a polynomial p such that Eq. (2) from Section 2.3 holds. We define a language L' as follows:

$$L' = \left\{ \langle x, w \rangle \;\middle|\; |w| = p(|x|) \wedge (\forall y \in \Sigma^{p(|x|)})(\exists z \in \Sigma^{p(|x|)})[R(x, w, y, z)] \right\}.$$

Clearly, $L' \in \Pi_2^{\mathrm{p}}$. By Theorem 4.3, there exists a number j such that $\mathrm{pad}'_j(L')$ is a P-k-king language. Let f be the tournament family specifier of the padded language.

We now describe a reduction from L to SUCCINCT-k-RADIUS. For an input word $x \in \Sigma^n$, let $n' = |\langle x, w \rangle|$ for some word $w \in \Sigma^{p(|x|)}$ (by Lemma 2.1, for a given x the number n' does not vary over the possible values of w). Consider the tournament T specified by f for the word length $m = n' + (n')^j + j + 3$. Let us write $p_{x,w}$ for $\mathrm{pad}_j(\langle x, w \rangle) = \langle x, w \rangle 0^{n'^j + j + 3}$. Then, by the definition of the padding function, we know that $\langle x, w \rangle \in L'$ holds if and only if $p_{x,w}$ is a k-king in T.

We define a graph G_x as follows. We add 2^m new vertices to the tournament T, let us call them σ_1 to σ_{2^m}. We add the following new edges to the graph: There is an edge from every $p_{x,w}$ to σ_1. There is an edge from each σ_i to σ_{i+1} for $i \in \{1, \ldots, k-2\}$. Finally, there is an edge from σ_{k-1} to all σ_i with $i > k-1$. The vertex set of G_x has size 2^{m+1}. It is easy to see that we can compute, in polynomial time, the code of a circuit C that specifies G_x.

All that remains to be shown is that G_x has radius k if and only if $x \in L$. To see this, first note that only the vertices $p_{x,w}$ can be k-kings of G_x: There is exactly one path of length $k-1$ leading to the vertex σ_k, namely the path $\sigma_1 \to \cdots \to \sigma_k$, and the only vertices from which we can reach σ_1 in turn are the vertices $p_{x,w}$. On the other hand, from each of these vertices we can clearly reach all σ_i in at most k steps. This implies that we can reach *all* vertices in G_x from $p_{x,w}$ within k steps if and only if $p_{x,w}$ is already a k-king of T. Thus, *some* $p_{x,w}$ is a k-king of G_x if and only if there exists a $w \in \Sigma^{p(|x|)}$ with $\langle x, w\rangle \in L'$. $\square$

6. Complexity of Top-Toda languages

In this section, we prove that the Top-Toda languages are in Π_2^{p}, but they cannot be NP-complete unless P = NP. Recall the definition of the Top-Toda languages: Given a commutative P-selector f, let $\text{Top-Toda}_f = \{x \in \{0,1\}^* \mid x$ is a k-king in the length-$|x|$ tournament specified by f for some $k\}$. Note that $\text{Top-Toda}_f = \bigcup_k k\text{-Kings}_f$.

Theorem 6.1. $\text{Top-Toda}_f \in \Pi_2^{\mathrm{p}}$ *for all tournament family specifiers* f.

Proof. Suppose we are given an input x, which corresponds to a vertex v of the length-$|x|$ tournament T specified by f, and must decide whether every vertex of this tournament is reachable from v. For this, we can use the observation from [12] that there is a Π_2^p-algorithm for the following problem: The input is a succinct representation of a tournament, a source, and a target vertex. It accepts if and only if there is path from the source to the target. For our Π_2^p-algorithm for Top-Toda_f we have to check whether for all vertices u of the tournament T there is a path from v to u. For deciding whether there is such a path, we use the Π_2^p-algorithm. So the overall complexity is $\forall \cdot \Pi_2^p = \Pi_2^p$ (we use the "$\forall$" operator here in its standard fashion), thus yielding a Π_2^p-algorithm for the overall problem. $\square$

Theorem 6.2. *If* Top-Toda_f *is* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-hard for* NP *for some tournament family specifier* f*, then* P = NP.

Proof. Suppose we could show that Top-Toda_f is P-selective. Selman [16] already noted that P-selective sets cannot be $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for NP unless P = NP. At first sight, the set Top-Toda_f appears to be P-selective: Given any two words x and y of the same length, apply f to them. If $f(x, y) = y$ and y is in the top Toda equivalence class, so is x (since every vertex reachable from y is also reachable from x in one more step). Unfortunately, the argument breaks down when the words x and y have different lengths. In this case, $f(x, y) = y$ does not imply that x must be in the top Toda equivalence class of the length-$|x|$ tournament if y is in the top Toda equivalence class of the length-$|y|$ tournament.

Although we cannot apply Selman's result directly, with a bit of extra effort we can adapt Selman's proof so that it also works for our situation. Let us call a language A *lengthwise P-selective* if the following holds: There is a polynomial-time computable function f such that for every two words x and y of the same length we have $f(x, y) \in \{x, y\}$ and $\{x, y\} \cap A \neq \emptyset$ implies $f(x, y) \in A$. Clearly, Top-Toda_f is lengthwise P-selective for every tournament specifier f. Thus, all that remains to prove is that if a lengthwise P-selective set is $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for NP, then P = NP.

Suppose A is lengthwise P-selective and suppose we can $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduce the satisfiability problem to A via some reduction machine R. We present a polynomial-time algorithm for the satisfiability problem: On input of a formula ϕ, we keep track of a list of formulas such that ϕ is satisfiable if and only if at least one formula in the list is satisfiable. Initially, the list just contains ϕ. We apply two operations to the list repeatedly: In an *expansion step*, we simultaneously replace each formula in the list that contains a variable by the two formulas obtained by substituting the variable once by true and once by false. Note that we do not violate the list invariant during the expansion step. After each expansion step we apply *pruning steps*, where we try to find two different formulas ρ and ψ in the list such that $R(\rho)$ and $R(\psi)$ have the same length. When we find such a pair, we apply the selector to $R(\rho)$ and $R(\psi)$. If the selector picks $R(\rho)$, we remove ψ from the list. If the selector picks $R(\psi)$, we remove ρ from the list. When pruning is no longer possible, we move on to another expansion step, followed by pruning once more, and so on.

Note that the pruning operation will never remove the only element from the list that is satisfiable. Thus, at the end, when neither expansion nor pruning is possible any more, the original formula ϕ will be satisfiable if and only if there is a true formula in the list.

It remains to argue that the algorithm runs in polynomial time. In each expansion step, the number of variables in every formula decreases by one, so there can only be as many expansion steps as there are variables in ϕ. So it suffices to show that the length of the list is never more than polynomial. For this, observe that there is a polynomial p such that the length of all list entries is at most $p(|\phi|)$ (and indeed, in most natural codings of formulas, each formula in the list would be of size less than or equal to the size of ϕ). Next, R can be time-bounded by some polynomial q. Then, all list elements are mapped by R to words of length at most $q(p(|\phi|))$. This means that whenever the length of the list exceeds $q(p(|\phi|))$, two different list elements are mapped to words of the same length and one word is pruned. This shows that the length of the list is at most $2 \cdot q(p(|\phi|))$, namely right after an expansion step. $\square$

7. Conclusion

In this article, we saw that king and k-king problems have tremendous flexibility, and in fact can be used as a naming scheme for the nontrivial Π_2^{p} many-one degrees. Using related techniques, we studied the complexity of radius, diameter, and Top-Toda problems. In the case of each of the king, k-king, radius, and diameter problems, we found that completeness held for classes at the second or third level of the polynomial hierarchy.

We discussed only directed graphs in the present article and it is natural to ask about the complexity of the diameter and radius problems for undirected graphs. It turns out that using arguments of a similar spirit to those presented in this article, one can show that (a) the succinct k-diameter problem for undirected graphs is still Π_2^{p}-complete for fixed $k \geq 2$ and (b) the succinct k-radius problem for undirected graphs is still Σ_3^{p}-complete for fixed $k \geq 2$; see [21] for detailed proofs.

Acknowledgements

We are very grateful to the anonymous referees for helpful comments and suggestions and to Felix Brandt for helpful discussions.

The first author was supported in part by grants NSF-CCR-0311021 and NSF-IIS-0713061, a TransCoop grant, and a Friedrich Wilhelm Bessel Research Award. Work done in part while on sabbatical at the University of Rochester and while visiting the Tokyo Institute of Technology.

The second author was supported in part by grants NSF-CCF-0426761 and NSF-CCF-0915792, JSPS Invitational Fellowship S-05022, a TransCoop grant, and a Friedrich Wilhelm Bessel Research Award. Work done in part while visiting the Tokyo Institute of Technology.

The fourth author was supported in part by ''New Horizons in Computing'' (2004–2006), an MEXT Grant-in-Aid for Scientific Research on Priority Areas.

References

[1] J. Bang-Jensen, G. Gutin, Generalizations of tournaments: A survey, Journal of Graph Theory 28 (4) (1998) 171–202.
[2] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Llull and Copeland voting computationally resist bribery and control, Journal of Artificial Intelligence Research 35 (2009) 275–341.
[3] H. Galperin, A. Wigderson, Succinct representations of graphs, Information and Control 56 (3) (1983) 183–198.
[4] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 1979.
[5] G. Gutin, The radii of n-partite tournaments, Mathematical Notes 40 (1986) 414–417.
[6] E. Hemaspaandra, L. Hemaspaandra, T. Tantau, O. Watanabe, On the complexity of kings. Technical Report URCS-TR905, Department of Computer Science, University of Rochester, Rochester, NY, USA, December 2006. Revised, November 2007.
[7] L. Hemaspaandra, C. Nasipak, K. Parkins, A note on linear-nondeterminism, linear-sized, Karp–Lipton advice for the P-selective sets, Journal of Universal Computer Science 4 (8) (1998) 670–674.
[8] L. Hemaspaandra, M. Ogihara, M. Zaki, M. Zimand, The complexity of finding top-Toda-equivalence-class members, Theory of Computing Systems 39 (5) (2006) 669–684.
[9] L. Hemaspaandra, L. Torenvliet, P-selectivity, immunity, and the power of one bit, in: Proceedings of the 32nd International Conference on Current Trends in Theory and Practice of Computer Science, in: Lecture Notes in Computer Science, vol. 3881, Springer-Verlag, 2006, pp. 323–331.
[10] H. Landau, On dominance relations and the structure of animal societies, III: The condition for score structure, Bulletin of Mathematical Biophysics 15 (2) (1953) 143–148.
[11] A. Meyer, L. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, in: Proceedings of the 13th IEEE Symposium on Switching and Automata Theory, IEEE Press, 1972, pp. 125–129.
[12] A. Nickelsen, T. Tantau, The complexity of finding paths in graphs with bounded independence number, SIAM Journal on Computing 34 (5) (2005) 1176–1195.
[13] V. Petrovic, C. Thomassen, Kings in k-partite tournaments, Discrete Mathematics 98 (1991) 237–238.
[14] M. Schaefer, C. Umans, Completeness in the polynomial-time hierarchy: Part I: A compendium, SIGACT News 33 (3) (2002).
[15] M. Schaefer, C. Umans, Completeness in the polynomial-time hierarchy: Part II, SIGACT News 33 (4) (2002).
[16] A. Selman, P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP, Mathematical Systems Theory 13 (1) (1979) 55–65.
[17] A. Selman, Some observations on NP real numbers and P-selective sets, Journal of Computer and System Sciences 23 (3) (1981) 326–332.
[18] A. Selman, Reductions on NP and P-selective sets, Theoretical Computer Science 19 (3) (1982) 287–304.
[19] L. Stockmeyer, The polynomial-time hierarchy, Theoretical Computer Science 3 (1) (1976) 1–22.
[20] L. Stockmeyer, A. Meyer, Word problems requiring exponential time, in: Proceedings of the 5th ACM Symposium on Theory of Computing, ACM Press, 1973, pp. 1–9.
[21] T. Tantau, Complexity of the undirected radius and diameter problems for succinctly represented graphs. Technical Report SIIM-TR-A-08-03, Universität zu Lübeck, Lübeck, Germany, 2008.
[22] D. West, Introduction to Graph Theory, 2nd edition, Prentice-Hall, 2001.
[23] C. Wrathall, Complete sets and the polynomial-time hierarchy, Theoretical Computer Science 3 (1) (1976) 23–33.